

Tìm hiểu khóa chính, khóa ngoại

Khoá chính (PRIMARY KEY) trong SQL

Ràng buộc PRIMARY KEY xác định duy nhất từng của từng dữ liệu trong bảng.

Khóa chính phải chứa giá trị DUY NHẤT và không được chứa giá trị NULL.

Một bảng chỉ có thể có MỘT khóa chính và trong bảng, khóa chính này có thể bao gồm một hoặc nhiều cột (trường).

Khoá ngoại (FOREIGN KEY) trong SQL

KEY FOREIGN là một khóa ngoại dùng để liên kết hai bảng với nhau.


KHÓA NGOẠI là một trường (hoặc tập hợp các trường) trong một bảng tham chiếu đến KHÓA CHÍNH trong một bảng khác.

Các chuẩn NF

1NF:

- Mỗi ô trong bảng nên chứa một giá trị đơn (không phải là một danh sách)
- Mỗi dòng (bản ghi) cần phải là duy nhất

Ví dụ 1NF

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION 
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3 rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

Bảng 1: Trong mẫu 1NF

Để xác định mỗi dòng là duy nhất, chúng ta nên xem xét thêm tới khóa chính Primary Key.

Quy tắc 2NF (Mẫu thường thứ hai)

- Quy tắc 1- Trở thành 1NF
- Quy tắc 2- Khóa chính cột đơn

Rõ ràng là chúng ta không thể tạo cơ sở dữ liệu đơn giản của mình ở dạng Chuẩn hóa thứ 2 trừ khi chúng ta phân vùng bảng ở trên.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 rd Street 34	Mr.
3	Robert Phil	5 th Avenue	Mr.

Bảng 1

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Bảng 2

Chúng tôi đã chia bảng 1NF của chúng tôi thành hai bảng: Bảng 1 và Bảng2. *Bảng 1 chứa thông tin thành viên. Bảng 2 chứa thông tin về các bộ phim được thuê.*

Chúng tôi đã giới thiệu một cột mới có tên là **Membership_id**, đây là khóa chính cho bảng 1. Các bản ghi có thể được xác định duy nhất trong Bảng 1 bằng cách sử dụng id thành viên

Quy tắc 3NF (Mẫu thường thứ ba)

- Quy tắc 1- Trở thành 2NF
- Quy tắc 2- Không có phụ thuộc chức năng bắc cầu

Để di chuyển bảng 2NF của chúng tôi thành 3NF, một lần nữa chúng ta cần chia lại bảng của mình.

Ví dụ 3NF

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3 rd Street 34	1
3	Robert Phil	5 th Avenue	1

BẢNG 1

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Bảng 2

SALUTATION ID	SALUTATION
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

Bảng 3

Inner Join

Inner join trong SQL là một cách để kết hợp dữ liệu từ hai hoặc nhiều bảng trong cơ sở dữ liệu quan hệ. Nó chỉ trả về những hàng có giá trị khớp trong cả hai bảng. Nói cách khác, inner join chỉ trả về các hàng có giá trị của các cột được chỉ định trong cả hai bảng khớp với nhau.

Cú pháp cơ bản cho phép nối bên trong như sau:

```
SELECT column1, column2, ...
```

```
FROM table1
```

```
JOIN table2
```

```
ON table1.column = table2.column;
```

Trong ví dụ này, table1 và table2 là tên của các bảng mà bạn muốn inner join, còn table1.column và table2.column là các cột mà bạn muốn khớp. Câu lệnh SELECT chỉ định cột nào bạn muốn truy xuất từ các bảng.

Bạn cũng có thể inner join nhiều bảng và cũng có thể sử dụng từ khóa "**JOIN**" thay vì "**INNER JOIN**" vì SQL hỗ trợ inner join ngầm định.

Left Join

`Left join` (còn được gọi là `left outer join`) trong SQL là một cách để kết hợp dữ liệu từ hai hoặc nhiều bảng trong cơ sở dữ liệu quan hệ, tương tự như `inner join`. Điểm khác biệt chính là `left join` trả về tất cả các hàng từ bảng bên trái (tức là bảng đầu tiên được chỉ định trong join), ngay cả khi không có kết quả khớp trong bảng bên phải (tức là bảng thứ hai được chỉ định trong join). Trong trường hợp khi tồn tại các record của `table1` không có kết quả khớp với `table2` thì các `column select` của `table2` sẽ được trả về bằng `NULL`.

Cú pháp cơ bản:

```
SELECT column1, column2, ...  
  
FROM table1  
  
LEFT JOIN table2  
  
ON table1.column = table2.column;
```

Trong ví dụ này, `table1` là bảng bên trái và `table2` là bảng bên phải, đồng thời `table1.column` và `table2.column` là các cột mà bạn muốn so khớp.

Right Join

`Right join` (còn được gọi là `right outer join`) trong SQL tương tự như `left join`, nhưng nó trả về tất cả các hàng từ bảng bên phải (tức là bảng thứ hai được chỉ định trong join), ngay cả khi không có kết quả khớp ở bảng bên trái (tức là bảng đầu tiên được chỉ định trong liên kết). Trong trường hợp khi tồn tại các record của **table2** không có kết quả khớp với **table1** thì các `column select` của **table1** sẽ được trả về bằng `NULL`.

Cú pháp cơ bản cho `right join` như sau:

```
SELECT column1, column2, ...  
  
FROM table1  
  
RIGHT JOIN table2  
  
ON table1.column = table2.column;
```

Full Outer Join

Full outer join (hoặc full join) trong SQL là sự kết hợp của cả left join và right join. Nó trả về tất cả các hàng từ cả hai bảng, ngay cả khi không có kết quả khớp trong bảng kia. Bảng kết quả sẽ chứa tất cả các hàng từ bảng bên trái, tất cả các hàng từ bảng bên phải và mọi hàng phù hợp từ cả hai bảng.

Full outer join không phải là thao tác SQL tiêu chuẩn, hệ thống cơ sở dữ liệu khác có thể hỗ trợ hoặc không hỗ trợ nó, đồng thời cú pháp về cách viết full outer join có thể khác nhau tùy theo cơ sở dữ liệu. Một số cơ sở dữ liệu sử dụng từ khóa FULL OUTER JOIN, một số sử dụng FULL JOIN.

Ví dụ, trong **MySQL**, bạn có thể mô phỏng FULL OUTER JOIN bằng cách sử dụng UNION của LEFT JOIN và RIGHT JOIN. Cú pháp cơ bản sẽ là:

```
SELECT column1, column2, ...  
FROM table1  
LEFT JOIN table2  
ON table1.column = table2.column  
UNION  
SELECT column1, column2, ...  
FROM table1  
RIGHT JOIN table2  
ON table1.column = table2.column;
```

Group by

Group by trong SQL là gì? Cách dùng Group by khi lập trình bằng SQL như thế nào? Hãy cùng Quantrimang.com tìm hiểu nhé!

Lệnh GROUP BY trong SQL Server được dùng để lấy dữ liệu tóm tắt dựa trên một hoặc nhiều nhóm. Các nhóm có thể được tạo thành từ một hoặc nhiều cột. Ví dụ, truy vấn GROUP BY sẽ được dùng để đếm số lượng nhân viên trong mỗi phòng ban, hoặc tính tổng lương của một bộ phận.

Bạn phải dùng các hàm tổng hợp như COUNT(), MAX(), MIN(), SUM(), AVG()... trong truy vấn SELECT. Kết quả câu lệnh GROUP BY trong SQL Server trả về một hàng cho từng giá trị của cột GROUP BY.

Cú pháp:

```
SELECT column1, column2,...columnN FROM table_name
```

```
WHERE
```

```
GROUP BY column1, column2...columnN
```

```
ORDER BY
```

Mệnh đề SELECT có thể bao gồm cột được dùng với SQL GROUP BY. Vì thế, để bao gồm các cột khác trong hàm GROUP BY, dùng các hàm tổng hợp như COUNT(), MAX(), MIN(), SUM(), AVG() với những cột đó.

Order by

Trong SQL, mệnh đề ORDER BY được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc theo thứ tự giảm dần trên một hoặc nhiều cột. Một số cơ sở dữ liệu mặc định sắp xếp các kết quả truy vấn theo thứ tự tăng dần.

Trong đó, lệnh ASC được sử dụng để sắp xếp tăng dần và DESC được sử dụng để sắp xếp giảm dần.

Cú pháp lệnh ORDER BY trong SQL

Cú pháp cơ bản của lệnh ORDER BY với câu lệnh SELECT sẽ như sau:

```
SELECT danh_sach_cot
```

```
FROM ten_bang
```

```
WHERE dieu_kien
```

```
ORDER BY cot1, cot2, .. cotN ASC --DESC;
```

Update

Cú pháp:

```
UPDATE table_name
```

```
SET column1 = value1, column2 = value2, ...
```

```
WHERE condition;
```